9

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/003,482	12/06/2001	Shmuel Ur	UR=1	4381

1444 7590 10/18/2005

BROWDY AND NEIMARK, P.L.L.C.
624 NINTH STREET, NW
SUITE 300
WASHINGTON, DC 20001-5303

EXAMINER	
RUTTEN, JAMES D	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 10/18/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/003,482	**Applicant(s)** UR ET AL.	
	Examiner J. Derek Rutten	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11,13-30 and 32-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11,13-30 and 32-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 August 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 5 August 2005 has been entered, wherein claims 1, 17, and 37-40 have been amended, claims 41 and 42 have been canceled, and no new claims have been added. Claims 1-11, 13-30, and 32-40 remain pending in the application and have been fully considered by the examiner.

Response to Arguments

2. Applicant's arguments in the last paragraph of page 21, with respect to the Spix reference as applied to claims 1 and 17 have been considered but are moot in view of the new ground(s) of rejection.

3. In the last sentence on page 21 through the third line of page 22, applicant suggests that neither IBM nor BOTI teach or suggest "generating a number of rules less than a number of basic blocks", nor that "the number of rules is a function of a control flow structure". However, new analysis of BOTI provides support for these limitations as detailed in the following rejection of claim 1. See BOTI page 11 line 33 – page 13 line 21. Thus, Applicant's argument is not convincing.

4. Applicant further argues in the third paragraph on page 22, that the cited art does not reveal "reduction factors in relation to numbers of basic blocks of the SUT", as related to a factor of a range of numbers less than the number of basic blocks. This argument is not convincing. It is noted that "reduction factors" are not recited in the rejected claims. However, in regard to the basic idea of generation of rules less than a number of basic blocks, BOTI teaches that optimizations using basic blocks as borrowed from compiler theory are used to reduce the state space of model checking tasks. See page 5 lines 10-19. BOTI continues on pages 12 and 13 with an example using Fig. 4 to teach that basic blocks in a control-flow structure for SUT 10 that contains 5 basic blocks can be used to solve a "subset cover problem." The solution of this subset cover problem produces a subset T which contains two blocks. The state space of the SUT 10 as depicted in Fig. 4 is thereby reduced from 5 basic blocks, to 2, which is a reduction by a factor of 2.5. BOTI thus provides an example for reducing the state space of a model checker.

Specification

5. The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code (page 1 line 19 and page 4 line 7). Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code. See MPEP § 608.01.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 32-34 and 37-40 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

8. Claims 32 and 33 recite the limitation "A method according to claim 31" in line 1. There is insufficient antecedent basis for this limitation in the claim. These claims will be interpreted as depending from claim 1. Claim 34 is rejected for being dependent upon a rejected base claim.

9. Claims 37 and 38 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. See MPEP § 2172.01. The omitted elements are: Any such hardware, device, or element that gives form to the "apparatus" as claimed. Although the claims are statutory "Machine" claims, the body of the claims does not support the preamble by setting forth a particular structure or relationship of the claim elements to any such apparatus. The elements of these claims are being interpreted as being performed by a computer processor in communication with system memory.

10. Claims 39 and 40 are rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential elements, such omission amounting to a gap between the elements. See MPEP § 2172.01. The omitted elements are: Any such encoding of instructions or other elements that gives form to the "product" as claimed. Although the claims are statutory "Manufacture" claims, the body of the claims does not support the preamble by setting forth a

particular structure or relationship of the claim elements to any such product. The elements of these claims are being interpreted as being performed according to computer instructions encoded on a computer readable medium.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1-10, 13-28, 30, 32-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over prior art of record "Coverability Analysis Using Symbolic Model Checking" published by IBM (hereinafter "IBM") in view of the "Background of the Invention" section found on pages 1-13 of the originally filed specification (hereinafter "BOTI").

In regard to claim 1, IBM discloses:

A method for performing coverability analysis in software, See IBM page 1:

> Every coverage model has a corresponding **coverability model**. A coverability model is defined by creating, for every coverage event indicator in coverage model, a coverability event indicator which is binary function on the state-machine model. The coverability event indicator is true if there exists a test on the state-machine model for which the corresponding coverage event indicator is true.

comprising:

formulating respective coverability tasks for the dominating blocks of the SUT;

See IBM bottom of page 1:

> First, as described above, a coverage model is in fact composed of coverage event indicators, each of which is **mappable to a corresponding coverability indicator**.

generating rules regarding behavior of the SUT corresponding respectively to the coverability tasks; See IBM bottom of page 1 – top of page 2:

> The second observation is that a state-machine model can be instrumented with control variables and related transitions which, on one hand, retain the original model behavior as reflected on the original state variables, and, on the other hand, can be used for coverability analysis of the model. The analysis is carried out by **formulating special rules** on the instrumented model, and presenting these rules (with the instrumented model) to a Symbolic Model Checker.

for each of the rules, running a symbolic model checker to test a behavioral model of the SUT, so as to produce respective results for the rules; See IBM top of page 2 as cited above:

> The analysis is carried out by formulating special rules on the instrumented model, and **presenting these rules (with the instrumented model) to a Symbolic Model Checker**.

and

computing a coverability metric for the SUT responsive to the results and the coverability tasks. See IBM top of page 1:

> ...it is shown how a number of **coverability metrics**, corresponding to some commonly-used coverage metrics (statement, multi-condition), **can be implemented** via Symbolic Model Checking (1).

wherein computing the coverability metric comprises:

evaluating an attained coverability responsive to the respective results produced by running the symbolic model checker; evaluating an unattained coverability responsive to the respective results produced by running the symbolic model checker; See page 1:

> A coverability model is defined by creating, for every coverage event indicator in coverage model, a coverability event indicator which is binary function on the state-machine model. The coverability event indicator is true if there exists a test on the state-machine model for which the corresponding coverage event indicator is true.

And further on page 2:

> The analysis is carried out by formulating special rules on the instrumented model, and presenting these rules (with the instrumented model) to a Symbolic Model Checker.

These passages show that rules are presented to a symbolic model checker, and an indicator function returns true or false depending on the return value of the binary function. In other words, an evaluation is made by the symbolic model checker to determine whether coverability has been attained or if coverability is unattained.

IBM does not expressly disclose *performing a static analysis of software under test (SUT) so as to identify a plurality of dominating blocks in the SUT,* comparison of attained coverability and coverability tasks, calculation based on the comparison, or analyzing the model based on unattained coverability.

However, in an analogous environment, BOTI teaches:

performing a static analysis of software under test (SUT) so as to identify a plurality of dominating blocks in the SUT (BOTI: page 11 line 11 – page 12 line6:

> As noted earlier, **some optimizations in model checking borrow concepts from compiler theory**. These concepts are known in the art, and include a basic block--a set of one or more statements within the same control-flow construct. Another useful, related concept is that of **dominating blocks**, including pre-dominating and post-dominating blocks.

Also Fig. 4 and associated text on page 12 lines 17-29 teaches dominating blocks.

performing a comparison between the attained coverability and the coverability tasks; See BOTI page 3 lines 14-15:

> The oracle function performs a comparison step 34 between actual results of execution 24 and expected results 32...

calculating the coverability metric responsive to the comparison; See page 3 lines 16-17:

> ...and condition 36 determines the success or failure of the test.

analyzing the behavioral model of the SUT with respect to the unattained coverability. See page 3 lines 17-19:

> An outcome of failure generally indicates a defect in SUT 10, which requires developer attention in a debug step 38.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of software test procedures with IBM's coverability tool. One of ordinary skill would have been motivated to analyze source code to identify dominating blocks in order to perform computational optimizations to reduce the amount of time spent analyzing a model. Further, one would be motivated to compute a relative success or failure of a test in order to determine whether further analysis is necessary.

Further, IBM does not disclose the generation of rules in relation to control-flow structures. However, BOTI teaches these limitations as follows:

Generating a number of rules less than, by a factor in a range from two to ten, a number of basic blocks in the SUT, and wherein the number of rules is a function of a control-flow structure of the SUT. BOTI teaches that optimizations using basic blocks are used to reduce the required work of a model checker. See page 5 lines 17-19. BOTI further teaches that a "greedy algorithm" presents a solution to a "subset cover problem" for a set of dominating blocks. See page 13 lines 3-21, especially lines 3-4, and 17-21:

> A subset cover problem, as is known in the art, may be solved on a set of dominating blocks.
>
> ...
>
> One such example is the Greedy Algorithm, which selects a block with the largest set of dominated blocks, constructs a list of covered blocks, and repeats the process until the list of covered blocks contains each block in the SUT.

Dominating blocks describe a relationship between basic blocks in a control-flow structure. See page 12 lines 1-6:

> These concepts are known in the art, and include a basic block - a set of one or more statements within the same control-flow construct. Another useful, related concept is that of dominating blocks, including pre-dominating and post-dominating blocks.

BOTI further describes the control-flow structure of Fig. 4 in relation to Table II and the subset cover problem on page 13. In this example, the subset cover problem is solved to produce a subset T:

> Solving the subset cover problem produces a subset T that covers all the basic blocks in SUT 10, i.e., if every basic block in subset T executes, all basic blocks in SUT must execute. By inspecting the preceding table, **it is noted that (B, C) comprise such a subset**, since, if Blocks B and C execute, Blocks A, D, and E must of necessity also execute.

The number of basic blocks in the subset T is less than, by a factor of 2.5, the number of basic blocks in SUT 10. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of the subset cover problem with IBM's teaching of a statement coverage/coverability model. The statement coverage/coverability model contains an event indicator for every statement. The subset cover analysis found in BOTI shows that only a subset of basic blocks requires such event indicators. Thus, the number of rules would be a function of the control-flow structure. One of ordinary skill would have been motivated to reduce the work required of a model checker (BOTI page 5 lines 17-19).

In regard to claim 2, the above rejection of claim 1 is incorporated. IBM does not expressly disclose: writing the SUT in a programming language adapted to define at least one of a group of elements comprising a software element and a hardware element.

However, BOTI teaches on page 4 lines 25-29 of the originally filed specification of the incorporated reference "Symbolic Model Checking without BDDs" by Biere et al. (hereinafter "Biere"). Further review of Biere reveals the use of the "SMV language" in Section 6. This leads to the reference "Symbolic Model Checking" by McMillan (hereinafter "McMillan") which defines the SMV language in Chapter 3. Since the SMV language is implemented as a software programming language, it inherently provides for software elements. McMillan then goes on to use the software elements in terms of hardware in Chapter 4. As such, it also defines hardware elements. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of SMV with IBM's model checker. One of ordinary skill would have been motivated to provide a symbolic description of the transition relation of a finite Kripke structure in order to provide a great deal of flexibility.

In regard to claim 3, the above rejection of claim 1 is incorporated. IBM does not expressly disclose: *wherein performing the static analysis of the SUT comprises: identifying a set of dominating blocks in the SUT; and solving a subset cover problem on the set of dominating blocks so as to identify the plurality of dominating blocks.* However, BOTI teaches solving a subset cover problem on page 13 lines 3-11. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of a subset cover problem with IBM's model checker. One of ordinary skill would have been motivated to use an efficient algorithm that solves the subset cover problem in order to save execution time.

In regard to claim 4, the above rejection of claim 3 is incorporated. IBM does not expressly disclose: *wherein the set of dominating blocks comprises a set of all dominating blocks in the SUT, and wherein the plurality of dominating blocks comprises fewer blocks than the set of all dominating blocks in the SUT.* However, BOTI teaches a subset of dominating blocks on page 13 line 5. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of a subset of dominating blocks with IBM's model checker. One of ordinary skill would have been motivated to reduce the computation space in order to reduce execution time.

In regard to claim 5, the above rejection of claim 4 is incorporated. IBM does not expressly disclose: *wherein running the symbolic model checker comprises performing a number of executions of the symbolic model checker smaller than a total number of all the dominating blocks in the SUT.* However, by the definition and example given in BOTI page 13 lines 17-21, the Greedy Algorithm "selects a block with the largest set of dominated blocks, constructs a list of covered blocks, and repeats until the list of covered blocks contains each block in the SUT." This results in a smaller number of "executions" than blocks. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of the Greedy Algorithm with IBM's model checker. One of ordinary skill would have been motivated to reduce the computation space in order to reduce execution time.

In regard to claim 6, the above rejection of claim 1 is incorporated. IBM further discloses: *wherein formulating the respective coverability tasks for the dominating blocks of the SUT comprises formulating coverability tasks by at least one of a group of methods* *comprising manual formulation and automatic formulation.* See IBM: "mappable to a corresponding coverability indicator." Mapping must be either manual or automatic, there are no there options.

In regard to claim 7, the above rejection of claim 1 is incorporated. IBM further discloses: *wherein generating the rules regarding behavior of the SUT comprises generating rules by at least one of a group of methods comprising manual generation and automatic generation.* See IBM: "formulating special rules on the instrumented model…". Formulation must be either manual or automatic, there are no other options.

In regard to claim 8, the above rejection of claim 1 is incorporated. IBM further discloses: *wherein running the symbolic model checker to test the behavioral model of the SUT comprises: evaluating the respective results so as to determine the truth or falsity of the rule; and generating a list of uncoverable elements responsive to the respective results.* See IBM: "The coverability event indicator is true if there exists a test on the state-machine model for which the corresponding coverage event indicator is true.

In regard to claim 9, the above rejection of claim 1 is incorporated. IBM further discloses: *wherein generating the rules regarding behavior of the SUT corresponding respectively to the coverability tasks comprises instrumenting the SUT by adding one or more statements and one or more auxiliary variables thereto, so as to facilitate evaluation of the rules.* IBM page 2: "formulating special rules on the instrumented model"; also "adding a counter after every statement and initializing it to zero."

In regard to claim 10, the above rejection of claim 9 is incorporated. IBM further discloses: *wherein instrumenting the SUT comprises: determining a plurality of basic* *blocks comprised in the SUT; and for each basic block: defining an auxiliary variable for the block; initializing the auxiliary variable to zero; and assigning the auxiliary variable a non-zero value upon execution of the basic block.* IBM page 2: "initializing it to zero... some of the counters are modified".

In regard to claim 13, the above rejection of claim 1 is incorporated. IBM further discloses: *analyzing a design of the SUT, responsive to the coverability metric, for at least one of a group of properties comprising dead code, unattainable states, uncoverable statements, uncoverable states, unattainable transitions, unattainable variable values, and unreachable conditions.* IBM page 2: "... a warning on the existence of **dead-code** is created for every statement that cannot be reached."

In regard to claim 14, IBM does not expressly disclose *applying a testing strategy chosen from one of a group of strategies comprising excluding uncoverable elements from coverage measurements, setting coverage goals responsive to the coverability metric, and determining a criterion for stopping testing responsive to the coverability metric.* However, BOTI teaches at least setting coverage goals on page 2 lines 3-29. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of coverage goals with IBM's coverability tool. One of ordinary skill would have been motivated to set coverage goals in order to attain a well-defined level of success.

In regard to claim 15, the above rejection of claim 14 is incorporated. IBM further discloses: *wherein the uncoverable elements comprise one or more elements chosen from a group of elements comprising uncoverable statements, uncoverable states,* *unattainable transitions, unattainable variable values, and unreachable conditions.* IBM page 1: "statement, multi-condition… define-use, mutation, and loop"; also page 2: "a warning on the existence of dead-code is created for every statement that cannot be reached."

In regard to claim 16, the above rejection of claim 1 is incorporated. IBM further discloses: *wherein formulating the respective coverability tasks for the dominating blocks of the SUT comprises: identifying a coverage model for the SUT; defining a coverability model for the SUT responsive to the coverage model; and generating the respective coverability tasks responsive to the coverability model.* IBM page 1: "a coverage model is in fact composed of coverage event indicators, each of which is mappable to a corresponding coverability indicator."

In regard to claim 17, IBM does not expressly disclose a second coverability task, an inflator, an inflated result, or evaluating a second coverability task responsive to the inflated result. However, BOTI teaches:

running a symbolic model checker comprising an inflator to test a behavioral model of the SUT responsive to the rule so as to produce an inflated result; See BOTI page 6 lines 26-29:

> Symbolic model checker system 56 contains an optional **inflator** 64 which expands the scope of the model checker output, as described in more detail below, with reference to FIG. 3

evaluating the second coverability task responsive to the inflated result. See BOTI page 7 lines 22-28:

> Inflator 64 provides a way to **include additional variables in the trace** in result 80, by generating plausible values for additional variables. Inflator 64 sets input variables to random values, and **computes values for additional values** based on the random input variables and the contents of the counter-example.

All further limitations have been addressed in the above rejection of claim 1.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of using an inflator with IBM's model checker. One of ordinary skill would have been motivated to introduce additional random variables into a system to overcome the variable space reduction introduced by the cone of influence optimization.

In regard to claim 18, the above rejection of claim 17 is incorporated. IBM does not expressly disclose: *wherein formulating the second coverability task comprises choosing a plurality of coverability tasks from a set of all coverability tasks for the SUT, and wherein evaluating the second coverability task comprises evaluating the plurality.* However, BOTI teaches on page 7 lines 25-28 that an inflator computes values based on random input variables and the contents of the counter-example. This result is fed back and executed until all coverable tasks have been examined. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of an inflator with IBM's model checker. One of ordinary skill would have been motivated to exhaust the computation space until all possible tasks have been evaluated.

In regard to claims 19, 21-28, 32-34, and 36, the above rejection of claim 17 is incorporated. All further limitations have been addressed in the above rejections of claims 3, 4, 5, 2, and 6-10, and 13-16, respectively.

In regard to claim 20, the above rejection of claim 19 is incorporated. IBM does not expressly disclose: *wherein selecting the first coverability task comprises: identifying a greatest-influence dominating block having a largest set of dominated blocks comprised in the plurality; and selecting the first coverability task responsive to the greatest-influence dominating block.* However, BOTI teaches the "Greedy Algorithm" on page 13 lines 17-21 for identifying optimal coverability tasks. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of the Greedy Algorithm with IBM's model checker. One of ordinary skill would have been motivated to reduce the computation space of the model in order to reduce execution time.

In regard to claim 30, the above rejection of 17 is incorporated. IBM does not expressly disclose: *wherein running the symbolic model checker comprises producing the inflated result regardless of the truth or falsity of the rule.* However, BOTI teaches inflation on page 7 lines 6-29, without regard to whether a rule is true or false. An inflator finds values outside of the cone of influence regardless of the value of any particular rule.

In regard to claim 35, the above rejection of claim 35 is incorporated. IBM does not expressly disclose: *performing a plurality of executions of an inflator program so as to produce a plurality of inflated results; and evaluating the second coverability task* *responsive to the plurality of inflated results.* However, BOTI teaches on page 7 lines 22-29 that an inflator is useful for obtaining a plurality of values outside the cone of influence. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use BOTI's teaching of inflators with IBM's model checker. One of ordinary skill would have been motivated to repeat the execution of an inflator in order to obtain additional results that lie outside the cone of influence.

13. Claims 11 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of IBM and BOTI as applied to claims 1 above, and further in view of U.S. Patent 5,579,515 to Hintz et al. (hereinafter "Hintz").

In regard to claim 11, the above rejection of claim 9 is incorporated. The combination of IBM and BOTI do not expressly disclose: *determining a plurality of basic blocks comprised in the SUT; defining a single auxiliary variable for the SUT; initializing the single auxiliary variable to zero; and assigning a unique non-zero value to the single auxiliary variable upon execution of each basic block.* However, in an analogous environment, Hintz teaches in column 3 lines 20-25 that a variable can be used to uniquely identify separate logical entities. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Hintz's teaching of unique non-zero entities in IBM's coverability tool. One of ordinary skill would have been motivated to uniquely identify an executed block in order to determine the coverage status of the block.

In regard to claim 29, the above rejection of claim 27 is incorporated. All further limitations have been addressed in the above rejection of claim 11.

14. Claims 37-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of IBM and BOTI, and further in view of U.S. Patent 6,484,134 to Hoskote (hereinafter "Hoskote").

In regard to claims 37 and 38, IBM does not expressly disclose an apparatus. However, in an analogous environment, Hoskote teaches such an apparatus in Fig. 1 and column 3 lines 18-57. All further limitations have been addressed in the above rejection of claims 1 and 17, respectively. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Hoskote's apparatus with IBM's method. One of ordinary skill would have been motivated to implement a method on an apparatus that can efficiently carry out the method.

In regard to claims 39 and 40, IBM does not expressly disclose a computer software product. However, Hoskote teaches such a product in claim 3 lines 48-64. All further limitations have been addressed in the above rejection of claims 1 and 17, respectively. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Hoskote's software product with IBM's method. One of ordinary skill would have been motivated to store instructions for a method for easy distribution and archival.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. "Efficient coverage testing using global dominator graphs" by Agrawal teaches the analysis of dominators in a control-flow graph for the purpose of reducing state space.

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571) 272-3703. The examiner can normally be reached on T-F 6:00 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TUAN DAM
SUPERVISORY PATENT EXAMINER

jdr